

Faculdade de Engenharia da Universidade do Porto



**Identificação e quantificação de células
oncocíticas em imagens microscópicas**

Jorge Afonso Nobre Costa

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
Major Automação

Supervisor: Rui Camacho (FEUP)

1st September 2014

A Dissertação intitulada

“Identificação e Quantificação de Células Oncocíticas em Imagens
Microscópicas”

foi aprovada em provas realizadas em 10-10-2014

o júri

Presidente Professora Doutora Maria Teresa Magalhães da Silva Pinto de Andrade
Professora Auxiliar do Departamento de Engenharia Eletrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto

Maria Teresa Pinto de Andrade

Professora Doutora Maria Benedita Campos Neves Malheiro
Professora Adjunto do Departamento de Engenharia Eletrotécnica do Instituto
Superior de Engenharia do Porto

Maria Benedita Malheiro

Professor Doutor Rui Carlos Camacho de Sousa Ferreira da Silva
Professor Associado do Departamento de Engenharia Informática da Faculdade de
Engenharia da Universidade do Porto

Rui Carlos Camacho de Sousa Ferreira da Silva

O autor declara que a presente dissertação (ou relatório de projeto) é da sua
exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente
autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou
inspirados em trabalhos de outros autores, e demais referências bibliográficas
usadas, são corretamente citados.

Jorge Afonso Nobre Costa

Autor - Jorge Afonso Nobre Costa

Faculdade de Engenharia da Universidade do Porto

© Jorge Costa, 2014

Resumo

Devido ao aumento da incidência de cancro na população e à necessidade de facilitar a identificação de células cancerígenas por parte de patologistas qualificados surgiu a ideia de criar uma ferramenta para ajudar os patologistas a detetar e contar células oncocíticas em imagens microscópicas, poupando assim trabalho. Desta forma, os médicos patologistas não teriam de ver todas as imagens, mas apenas as que fossem consideradas críticas (com possível tumor), pois muitas das imagens de pacientes são imagens sem qualquer patologia associada. Esta tese foca-se no trabalho de criação de um software que ajude os patologistas a detetar mais facilmente possíveis cancros ou células cancerígenas. Para esse efeito foram selecionadas imagens disponíveis na base de dados *The Cancer Genome Atlas* (TCGA), processadas em três fases: na primeira fase foram convertidas e divididas para que o pré-processamento fosse mais célere e porque as imagens inteiras são muito grandes; na segunda parte as imagens foram pré-processadas, selecionando-as e convertendo-as para tons de cinza para posteriormente serem aplicadas funções para se extraírem os contornos dos objetos. Como etapa final, extraíram-se as características dos objetos identificados como células pelo patologista para poderem ser utilizadas ferramentas de *machine learning*. As técnicas de *machine learning* têm como objetivo a identificação automática das células, discriminando células cancerígenas das não cancerígenas.

Abstract

Due to the increased incidence of cancer in the population and the need to facilitate the identification of cancer cells by machines, this thesis main goal is to create a tool that helps pathologists to detect and count these cells in microscopic images, thus saving time and work. Therefore, the pathologists would not need to see all of the images but only those that were considered to have possible cancer cells by the software, because many of the images are images of patients without any associated pathology. This thesis focuses on the creation of a software that helps pathologists to easily detect potential cancers or cancer cells. For this purpose, some images were selected from the data base TCGA and submitted to three phases: in the first phase they were converted and divided so that the preprocessing was faster and because the whole images were too large; in the second phase the images were preprocessed, selecting them and converting them to grayscale to extract the contours of the objects. As a final step, extracting characteristics of the identified objects and having a pathologist to confirm the cells in order to be submitted to machine learning tools. The use of machine learning enables the automatic differentiation between cancer cells and non-cancer cells.

Acknowledgements

To Joana for being there all the time for me and put up with my stress despite not being there.

To my family and all my friends for all the effort and support they gave to me, to accomplish this goal in my life.

To my supervisor for the help on this work.

Índice

Resumo	iii
Abstract.....	v
Acknowledgements	vii
Índice.....	ix
List of figures	xii
Abbreviations	xv
Chapter 1.....	1
Introduction.....	1
1.1 - Objectives.....	1
1.2 - Organization	2
Chapter 2.....	3
State of the art.....	3
2.1 - Biology of the cell	3
2.2 - Image processing.....	8
2.3 - Data mining	9
2.4 - Related software.....	17
Chapter 3.....	18
A tool for oncocyte cell identification	18
3.1- Development	19
3.2 - Results	22
Chapter 4.....	28
Machine learning	28
4.1 - Development.....	28
4.2 - Results	29
Chapter 5.....	31
Conclusion and future work.....	31
5.1 - Conclusion.....	31
5.2 - Future work.....	31

References..... 33

List of figures

Figure 1 -Different organelles present in an animal cell (from Encyclopedia Britannica, Inc)4

Figure 2 - Representation of the composition of Mitochondria (from Nature Education1).....5

Figure 3 - Normal Cell cycle. There are different checkpoints that the cells need to pass in order to guarantee that the cell is normally divided into two daughter cells. When these checkpoints are not properly controlled, the cells can divide indefinitely and lead to formation of tumours (adapted from [5])..... 6

Figure 4 - Example of a papillary carcinoma composed of oncocytic cells that show a pronounced chronic inflammatory cell infiltrate (from [8])7

Figure 5 - The life cycle of CRISP-DM (from [15]) 11

Figure 6 - Example of decision tree to get credit 13

Figure 7 - Bayes theorem 14

Figure 8 - kappa formula, Pr(a) represents observed accuracy and Pr(e) expected accuracy 15

Figure 9 - F-measure formula 15

Figure 10 - Examples of images from the TCGA data base..... 19

Figure 11 - Gray level transformation function for high contrast enhancement 21

Figure 12 - Image from the TCGA data base with the reference “TCGA-BQ-7053-01Z-00-DX1” 23

Figure 13 - Magnified part of the Figure 12..... 24

Figure 14 - Result of converting Figure 13 to gray scale 25

Figure 15 - Result of applying high contrast functions to Figure 14 26

Figure 16 - Binary image from the possible cells 26

Figure 17 - Borders of the objects found in Figure 16 27

Figure 18 - Results of the classification with 5 attributes 29

Figure 19 - Results of the classification with 46 attributes 30

Abbreviations

2D	2 Dimensional
ATP	Adenosine Triphosphate
BIOCAT	BIOimage classification and Annotation Tool
CNR	CellNote Results
CRISP-DM	Cross-Industry Standard Process for Data Mining
CSV	Comma Separated Values
DNA	Deoxyribonucleic Acid
FCUP	Faculdade de Ciências da Universidade do Porto
FNAC	Fine-Needle Aspiration Cytology
GNU	GNU is Not Unix
GPL	General Public License
IBMC	Instituto de Biologia Molecular e Celular
ILP	Inductive Logic Programming
IP	Image Processing
IPATIMUP	Institute of Molecular Pathology and Immunology at the University of Porto
JPEG	Joint Photographic Experts Group
KDD	Knowledge Discovery in Databases
ML	Machine Learning
NIH	National Institute of Health
PNG	Portable Network Graphics
SQL	Structured Query Language
SVM	Support Vector Machine
SVS	ScanScope Virtual Slides
TCGA	The Cancer Genome Atlas
TIFF	Tagged Image File Format
UBI	Universidade da Beira Interior
UP	Universidade do Porto

Chapter 1

Introduction

Certain types of cancer (such as thyroid and kidney) often show cells that have abnormally large numbers of mitochondria. These oncocytic tumor cells are larger, have a rounder nucleus, lower intercellular space and enhanced eosinophil staining. At Instituto de Patologia e Imunologia da Universidade do Porto (IPATIMUP) they have shown that the identification and quantification of these cells are important in terms of prognosis and selection of therapy [1], being associated with a mutational pattern more pathological [2]. However, there are few pathologists in the world that can properly identify this phenotype when making a histological diagnosis. In this work we will develop a software tool to identify and quantify the oncocytic cancer cells in high resolution microscopic images. As a test, we will use it on the online database provided by the National Institute of Health (NIH) for a variety of tumors in various organs, containing histological images of great quality [3].

This work will have a great impact in cancer diagnosis, because it will help pathologists to identify oncocytic cancer cells and save valuable time.

We now state the objectives of the work and the organization of this thesis.

1.1 - Objectives

This thesis aim is to be able to detect objects in images by applying algorithms developed specifically for that task and create a software that detects automatically cancer cells in high resolution microscopic image.

The objectives will be achieved with the following procedures:

- Transform the images from .svs to .tiff;
- Detect the outlines of objects present in the images;
- Extract characteristics from the objects;
- Use machine learning algorithms to identify cancer cells automatically;

The thesis work will contribute for the scientific community by giving a tool that selects images with potential cancer and discards some that do not have a presence of tumors cells and, in this way, saving time to the pathologists.

1.2 - Organization

The thesis is organized as follows.

Introduction - presents the problem of this thesis.

State of the art - gives an overview of all the concepts present on the thesis.

A tool for oncocyte identification - explains procedures and presents the results of the processes to create a tool for the identification of oncocyte cells.

Machine learning - presents the machine learning algorithms and their results.

Conclusion ad future work - describes the conclusions and the possible work that can be done in the future.

Chapter 2

State of the art

In this chapter, we survey the main topics included in the thesis concerning basic concepts in cell Biology. Furthermore, some beginner techniques and tools/libraries applied in image processing and machine learning will be explained in detail. Finally, we will do an overview of the programs that are currently used in bioimaging.

2.1 - Biology of the cell

A cell is the structural, functional and fundamental unit of all living organisms. The human body is composed of a huge amount of cells. They provide structure for the body, take in nutrients from food, convert those nutrients into energy and carry out specialized functions.

A cell is a chemical system that is able to maintain its structure and reproduce and it is the smallest unit of matter that can carry on all the processes of life [4].

Internal organization of cells

Cells contain a variety of internal structures called organelles. Organelles are specialized structures that perform certain tasks within the cell (**Figure 1**). Human cells contain several organelles in the Cytoplasm, namely the Cytoskeleton (structure that gives the cell its shape, capacity to move, and ability to arrange organelles and transport them from one part of the cell to another), Golgi apparatus (packages proteins inside the cell before they are sent to the final destination), Nucleus (contains most of the cell DNA, enclosed by a double layer of membrane), Plasma membrane (membrane that separates the interior of all cells from the outside), Ribosomes (large and complex molecular machinery, found within all living cells, that functions as the primary site of protein synthesis) and Mitochondria [4].

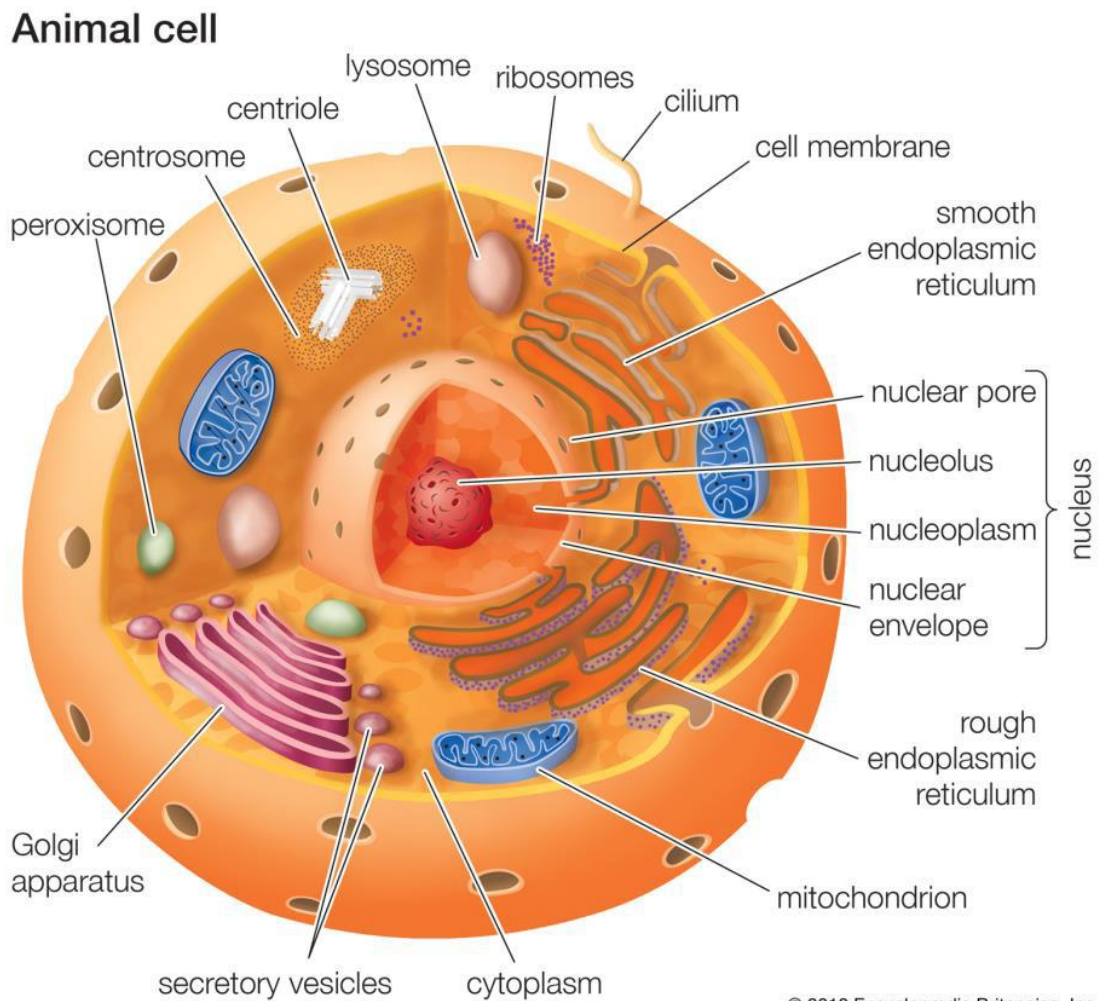


Figure 1 - Different organelles present in an animal cell (from Encyclopedia Britannica, Inc)

Mitochondria are organelles that are surrounded by a double membrane, retain their own small genome and provide the energy for all the movements of the cell.

Mitochondria resemble bacteria in some aspects as size and shape. They contain DNA, produce proteins and reproduce by dividing in two. Mitochondria are responsible for cellular respiration. The chemical energy produced by the mitochondria is stored in a small molecule called Adenosine Triphosphate (ATP). The cristae¹ greatly increase the inner membrane's surface area. It is on these cristae (Figure 2) that food (sugar) is combined with oxygen to produce ATP - the primary energy source for the cell [4].

¹ The cristae give the inner mitochondrial membrane its characteristic wrinkled shape providing a large amount of surface area for chemical reactions to occur on.

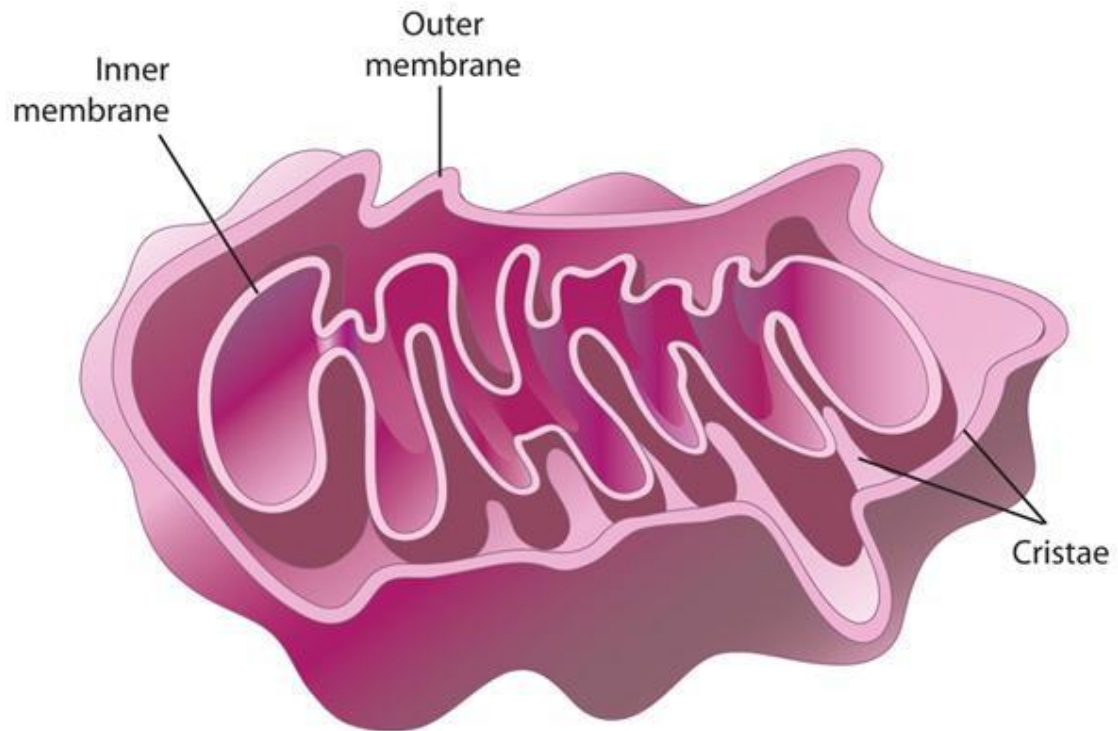


Figure 2 - Representation of the composition of Mitochondria (from Nature Education¹)

Division of the cells

Normal cells grow and divide in a normal fashion, in accordance with the cell cycle (**Figure 3**). Mutations in proto-oncogenes (genes that normally control how often a certain cell divides) or in tumor suppressor genes allow a cancerous cell to grow and divide without the normal controls imposed by the cell cycle, leading to the formation of tumours [5].

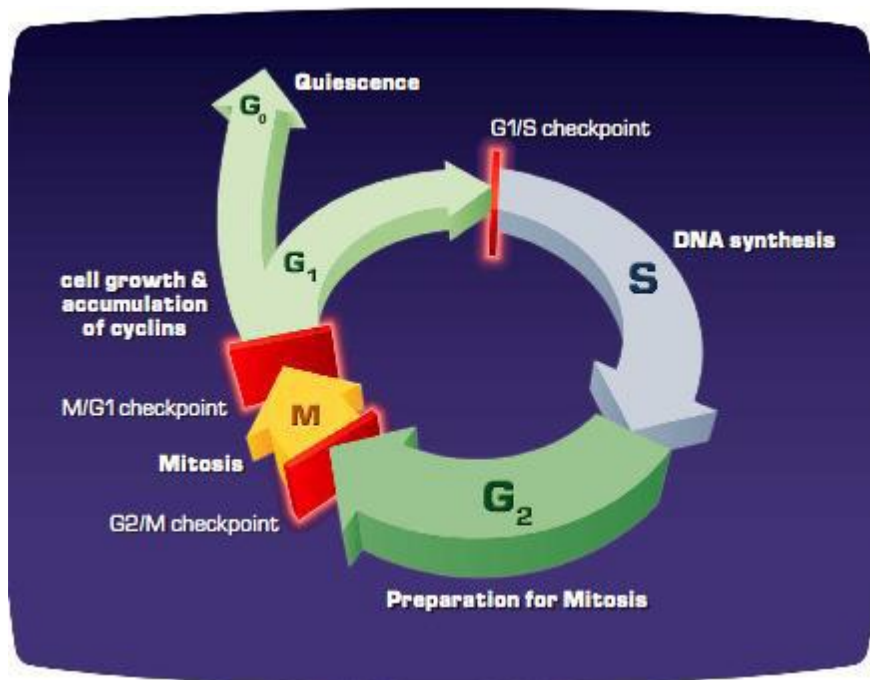


Figure 3 - Normal Cell cycle. There are different checkpoints that the cells need to pass in order to guarantee that the cell is normally divided into two daughter cells. When these checkpoints are not properly controlled, the cells can divide indefinitely and lead to formation of tumours (adapted from [5])

Oncocytic cells

Oncocytes are epithelial cells with granular, large and eosinophilic² cytoplasm, a central pyknotic³ nucleus, and, ultra-structurally, are composed of an unusual number of mitochondria of various sizes [6,7].

The proliferation of oncocytes leads to hyperplastic and neoplastic nodules [8](**Figure 4**).

Oncocytic tumours are found in several organs like in the salivary glands, thyroid, pituitary and in organs like pancreas, liver, lung, gut and kidney [6,7,8].

Oncocytes are cells which can be seen in a different number of conditions ranging from hyperplastic changes to malignant conditions, leading to difficulties in the diagnosis. They are thought of as metaplastic cells formed in response to adverse changes, with the normal cells losing their original specialization [7].

² **Eosinophilic** refers to the staining of certain cells, tissues or organelles after they have been washed by a dye called eosin.

³ **Pyknotic** refers irreversible condensation of chromatin in the nucleus of a cell undergoing programmed cell death or apoptosis.

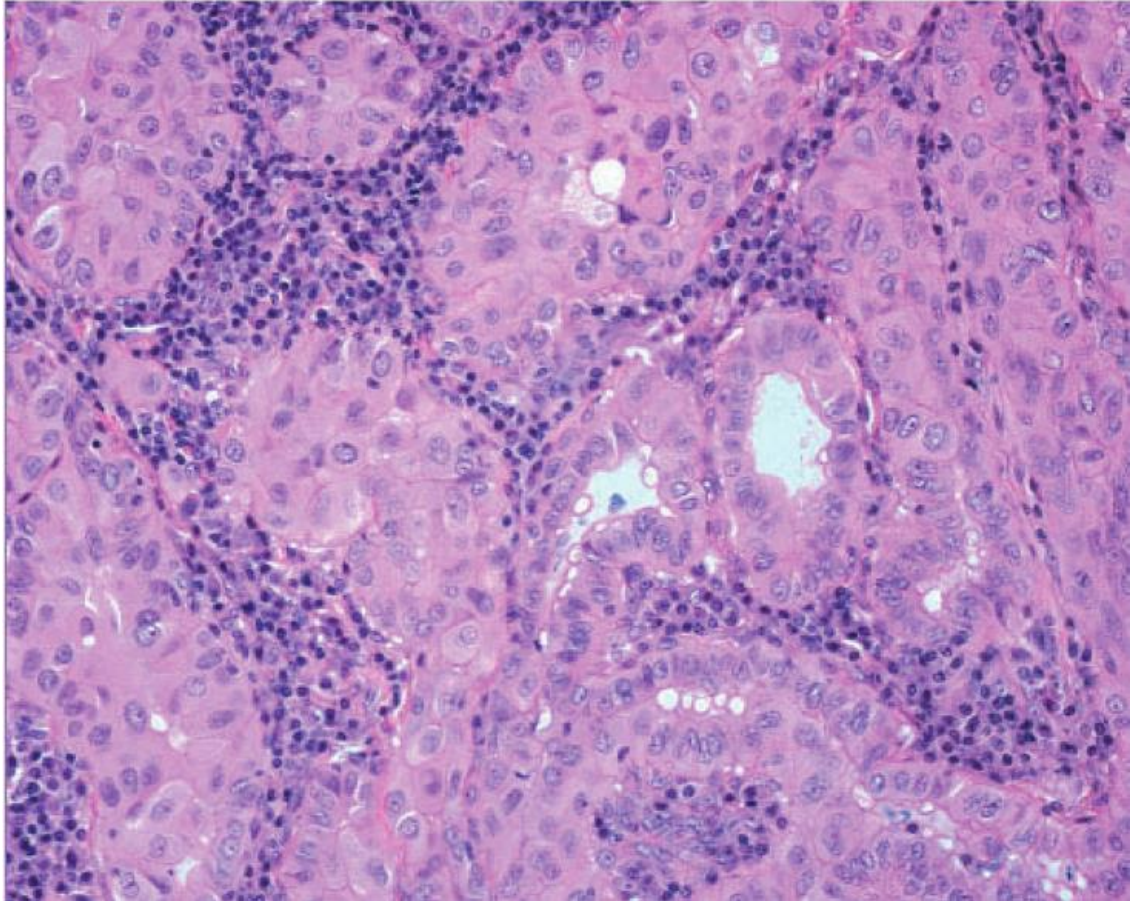


Figure 4 - Example of a papillary carcinoma composed of oncocytic cells that show a pronounced chronic inflammatory cell infiltrate. (from [8])

Available techniques in the diagnosis

The difference between a benign and a malignant neoplasm is based on cytological analysis of Fine-Needle Aspiration Cytology (FNAC) that can be very difficult due to focal sampling of the lesion, as oncocytic changes can occur in several neoplastic as well as non-neoplastic conditions. Histopathology remains the gold standard technique to reach the precise diagnosis [8].

Moreover, histopathology combined with image processing and analysis can be a great advantage in order to reach a more precise diagnosis.

2.2 - Image processing

Image processing is a research area of computer science that started in the 1970s and had a significant growth in the last two decades. There is a high number of applications where Image Processing (IP) can be used such as on autonomous guidance, automatic surveillance, medical imaging and diagnosis (Bio imaging). Nowadays, there are plenty of software and hardware for a large range of purposes and the interest on the topic is high.

The signal processing of any kind of image that results in another image or a set of measurements associated to that image is known as image processing. It is used by applying normal signal processing procedures to two dimensional (2D) signals.

We next explain the procedures typically used in image processing and, then, some existing useful libraries and algorithms will be described.

Algorithms

One image can be represented by a matrix with two dimensions. The height and width represent the location and a third value represents the color or intensity of the pixel. Once we have an image encoded as a matrix we can apply some algorithms to process, analyze and understand the image.

To process the image some punctual, local or global operations can be performed. Transforming the image into its negative, applying a threshold and enhancing the brightness or contrast are examples of algorithms applied punctually. Procedures locally applied result in a transformation using not only the pixel of the result but also its neighborhood. It can be used to smooth the image, to enhance brightness differences, to erode or dilate some objects and to detect edges. There are transformations where every pixel in the final result is affected by all the pixels from the image prior to the transformation, also called global procedures. Compression and rotation of the image are examples of these operations [9].

Libraries

There is a large variety of open source libraries in C++ to process and analyze images. In the rest of this subsection we list and describe some of the most popular in IP, pointing out their strengths and weaknesses.

The CImg library [10] stands for cool image. It is an efficient, easy to use library and distributed under the CeCILL-C or CeCILL licenses (close to GNU LGPL and GPL). It has a few classes defined and others can be added as plug-ins from other libraries like OpenCV and

The state of the art

libpng. It runs on every platform and can be compiled by numerous compilers (g++,icc, Visual C++, etc) and read different types of images (.png,.tiff,.jpeg,etc).

Libpng [11] it is an open source library focused on images in the Portable Network Graphics (PNG) format. It is tested for over 18 years and is released under the libpng license.

OpenCV [12] stands for Open Source Computer Vision Library and it includes hundreds of computer vision algorithms. The library is divided in modules. The core of the library (imgproc) has the usual functions for linear and non-linear images (transformations, histograms, etc), for video has video analysis functions such as background subtraction and object tracking and objdetect that detects defined classes (faces, mugs, people, etc). It uses BSD-license and has C, C++, Python and Java interfaces, supporting Windows, Mac Os, iOS, Android and Linux.

Camellia [13] is an easy to use open source image processing library that complements OpenCV library. Is written in C language and supports Windows and Linux. Has some exclusive algorithms and optimizations and supports 1, 8 and 16-bits images. The main features include color conversion, warping, drawing, labeling and filtering.

OpenCV will be the library used on this problem because has the largest compilation of algorithms and it was used in other programs alike.

2.3 - Data mining

KDD process

The process of extracting information from large quantities of data is named Knowledge Discovery in Databases (KDD) or data mining [14]. It has the following steps:

- **Data Cleaning** eliminates noise and unrelated data;
- **Data Integration** conglomerates various sources of data;
- **Data Selection** selects important data from database;
- **Data Transformation** applies summary or aggregation procedures to alter the data;
- **Data Mining** extracts data patterns using intelligent techniques;
- **Pattern Evaluation** recognizes the truly relevant patterns for the problem;
- **Knowledge Presentation** presents the information acquired to the user.

CRISP-DM process

Cross-Industry Standard Process for Data Mining (CRISP-DM) it is a DM model that is commonly used by expert data miners and comprises a cycle (**Figure 5**) that breaks the process in six stages:

- **Business Understanding** focus on the objectives and requisites from a business perspective, then converting it into a DM problem definition and a plan to achieve the objectives;
- **Data Understanding** focus on the familiarization with the data, collecting it and the proceeding activities. It makes easier to identify quality problems, notices first insights into data or to discover interesting subsets to form hypotheses;
- **Data Preparation** covers all the tasks done in the initial data to build the final dataset. It includes record, table, attribute selection, cleaning and transformation tasks;
- **Modeling** it is selected and are applied various modeling techniques;
- **Evaluation** evaluates thoroughly the model or models that have a high quality from a data analysis perspective. At the end, a decision on the use of DM results should be reached;
- **Deployment** is the organization and presentation of the results in a way that the client can use.

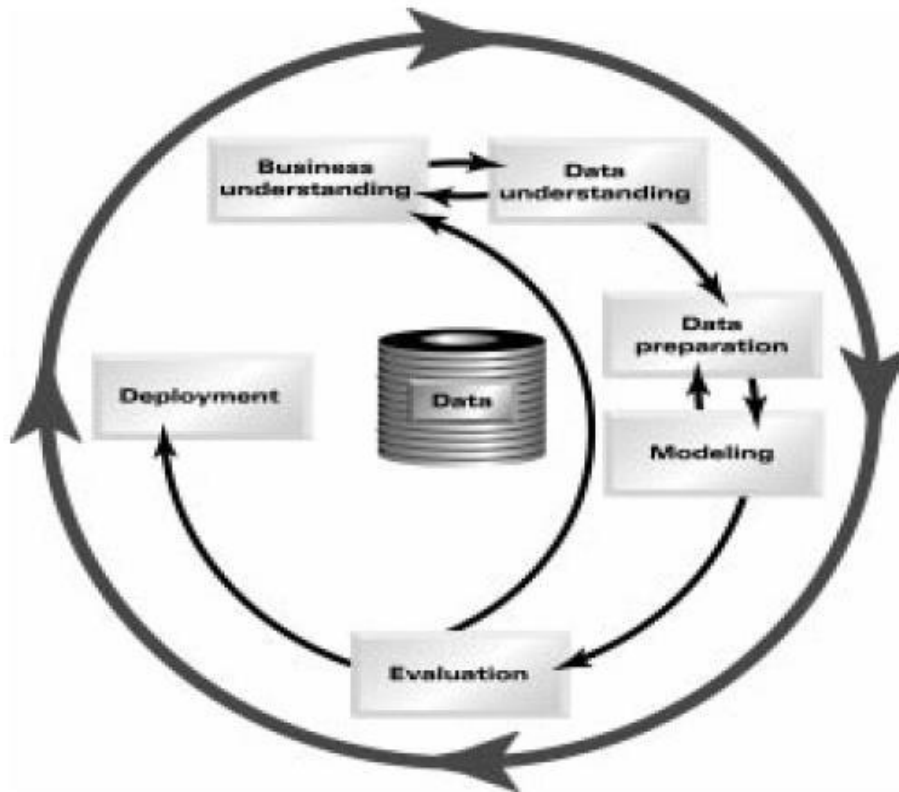


Figure 5. The life cycle of CRISP-DM (from [15])

The tasks can be divided in two different categories: descriptive and predictive. The first is to characterize the general properties of the data in the database and the second one is to perform inference on the current data in order to make predictions.

So, there are different data mining functionalities with different applications.

Concept description: Characterization and Discrimination

It can be useful to describe individual classes and concepts in concise and precise terms. Characterization is a summarization of the general features of the target class (class in study). Outputs can be pie charts, curves, bar charts, etc. Discrimination is when a target class is compared with other contrasting classes. The outputs are similar to the ones from characterization but should have comparative measures to distinguish between the classes.

Association analysis

Association rules can be used to associate some attributes that occur frequently together in a given set of data. They are commonly used in marketing and transaction data analysis.

Classification and Prediction

The process of finding a set of functions that could describe and distinguish different types of data classes or concepts. They can be represented with classification rules, decision trees, neural networks, etc. Classification is often preceded by relevance analyses that exclude attributes that don't contribute to the classification.

Clustering analysis

Clustering is used to analyze data objects without consulting a known class label from classification. It tries to maximize the similarities within the same cluster and minimize between different clusters, being a good way to form classes if they are not known by the user.

Evolution and deviation analysis

Evolution analysis describes and models regularities or trends for objects whose behavior changes over time. In time related data analysis it is often desirable not only to model the evolution but also the deviations from the trends which occur over time. They are the differences between measured values and corresponding references, such as previous values or normative values [14].

DM Algorithms

Decision Trees

Decision trees [16] are a decision-making algorithm that uses a tree-diagram to classify based on decisions and their possible consequences. Being a visual algorithm they are easily to comprehend and allow an overall view of a potentially complex situation. They can be linearized into classification rules where the outcome is the content of the leaf node. An example is show in **Figure 6**.

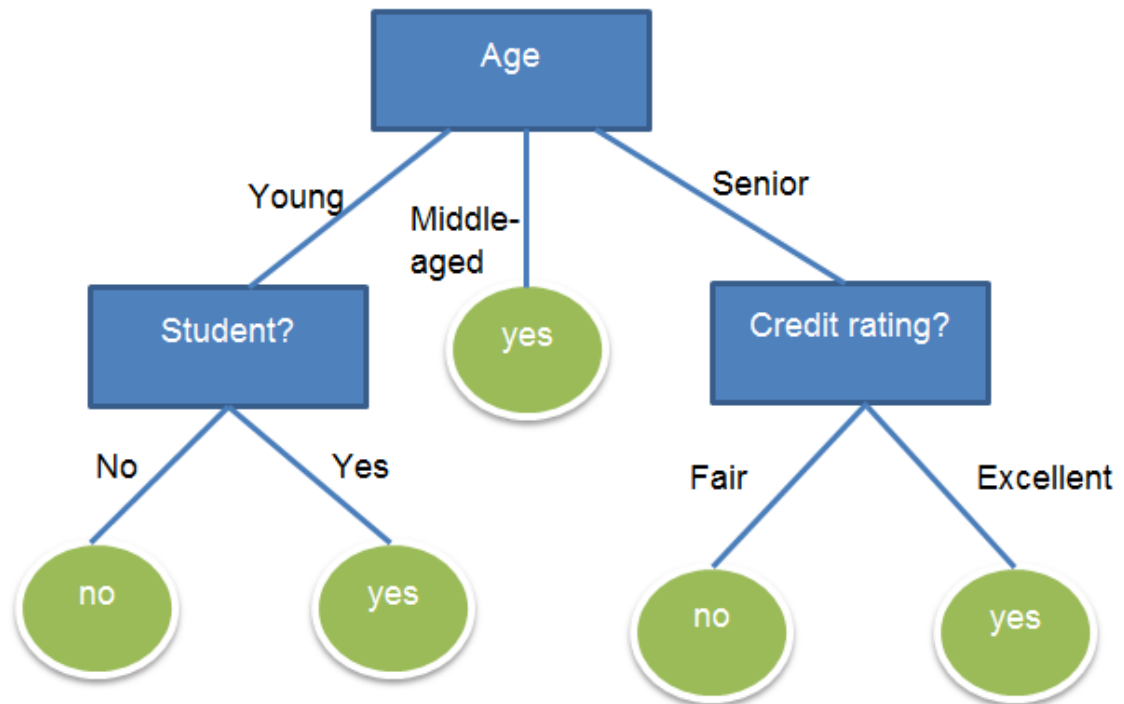


Figure 6. Example of decision tree to get credit.

Classification via Clustering

Clustering can be used as a classification algorithm. It generates clusters based on the different characteristics and then associates the clusters to previous labeled classes. If there are clusters classified as being from the same class, they are added.

SVM

Support Vector Machines (SVM) is a classification and regression method that can generate nonlinear decision boundaries using methods for linear classifiers and uses kernel functions that allow the user to classify data that have no obvious fixed-dimensional vector space representation (such as DNA sequence). With a set of examples divided in two categories, an SVM training algorithm creates a model that assigns the future data into one of two categories: it can be represented as points in space divided by a line that also divides the categories [17].

K-nearest Neighbor

K-nearest Neighbors algorithm is used in classification and regression. The input for the algorithm consists of the k closest training data in the feature space and the output is a class

membership. The new data is classified by a majority vote of their neighbors. It is the simplest machine learning algorithm [18].

Decorate

Decorate is a meta-learner for building diverse ensembles of classifiers by using specially constructed artificial training examples. Comprehensive experiments have demonstrated that this technique is consistently more accurate than the base classifier, Bagging and Random Forests [19].

Naive Bayes

Naive Bayes is a simple probabilistic classifier based on applying Bayes' theorem (represented on **Figure 7**) with strong (naive) independence assumptions [20].

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}.$$

Figure 7. Bayes theorem

In simple terms, this classifier assumes that the value of a particular feature is not related to the presence or absence of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 3" in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of the presence or absence of the other features.

An advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Performance Measures

Accuracy

Accuracy is the percentage of instances correctly classified by the method used.

Kappa

Kappa statistic is a metric that compares an observed accuracy with an expected accuracy (random chance) represented on **Figure 8**. It is less misleading than simply accuracy as a metric because it takes random chance into account. Landis and Koch [21] considers kappa values of 0-0.20 as slight, 0.21-0.40 as fair, 0.41-0.60 as moderate, 0.61-0.80 as substantial and 0.81-1 as almost perfect.

$$\kappa = \frac{\text{Pr}(a) - \text{Pr}(e)}{1 - \text{Pr}(e)},$$

Figure 8. kappa formula, $\text{Pr}(a)$ represents observed accuracy and $\text{Pr}(e)$ expected accuracy

Precision and recall

Precision, also known as positive predictive value, is a metric that compares the number of classified instances with correct classified instances. And recall, also known as sensitivity, is a metric that compares the number of classified instances with the real number of instances labeled with the same class.

F-Measure

F-measure is a metric measure that represents the harmonic mean of precision and recall represented on **Figure 9**.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Figure 9. F-measure formula

DM Tools

Weka

Weka [22] is an open source data mining and statistical library that uses GNU GPL and is written in Java. Its main strengths lie in the classification area, where all current ML approaches have been implemented within a clean, object-oriented Java class hierarchy. Supports several typical KDD tasks, particularly data preprocessing, clustering, classification, regression, visualization and feature selection. Weka provides access to SQL databases

utilizing Java Database Connectivity and can process the result returned by a database query. Its main user interface is the Explorer, but the same functionality can be accessed from the command line or through the component-based Knowledge Flow interface. This library can work with weka files (.arff) and Comma Separated Values (CSV) files.

RapidMiner

RapidMiner [23], formerly called YALE, is an environment for machine learning and data mining. It is the most used program to research and data mining tasks. It uses learning schemes and attributes evaluators from the Weka library. RapidMiner is divided in two different tools, one for data-analysis and one engine to data mining. It runs in all of the operating systems.

KNIME

KNIME [24] is open source software very easy to use and learn. It is written in Java, based on the software Eclipse and runs in Windows, Mac OS X and Linux. Users can install plugins for image, text processing and integrate other open source projects, such as Weka or R programming language. Allows the user to visually create data pipelines and execute all or some analysis and then present the results or models formed.

Orange

Orange [25] is open source data mining and machine learning software that features powerful, fast, friendly and versatile visual programming for data analysis and visualization. It already contains a large number of techniques displayed in forms of widgets (filters, models, models evaluation, etc). It includes algorithms for classification, such as majority, naive Bayes classification, neural network, classification trees, for clustering using *K*-means, hierarchical and consensus clustering. It's written in C++ and Python language and is based on Qt framework. It is a tool really easy to learn to work with and can run in Windows, Mac OS X and variety of Linux operating systems. It can read data from a large variety of file extensions (.tab;.data;.rda;.arff;.svm;.xml;.R;.cvs).

The data mining tool selected to be used was the Weka because has all the algorithms that will be used and was suggested by the supervisor.

2.4 - Related software

There are some computer applications/software in the market that are being used to solve the problems in Biology using bioimage processing. Following, we give some examples as well as describe these applications, their pros and cons.

BIOCAT

BIOimage Classification and Annotation Tool (BIOCAT) [26] was developed to recognize patterns in 2D or 3D bio images. It is able to extract, select and classify images with a range of 20 algorithms. The processes are made in modules and can be aligned in various ways that make the software a versatile tool to solve numerous problems. It has open access and has the ability to include new contents through plugins. It is a user-friendly program and can be customized in several ways to solve efficiently and effectively a grand variety of biological difficulties involving image classification.

CellNote

CellNote [27] is a software invented by Instituto de Biologia Molecular e Celular (IBMC) in collaboration with Faculdade de Ciências da Universidade do Porto (FCUP), Universidade do Porto (UP) and Universidade da Beira Interior (UBI) where the user can annotate different types of objects (cells) and their subgroups. This software supports numerous types of image formats, and single RGB channel can be selected. The software quantifies and presents different types of data and generates CellNote Results (CNR) file that can be exported to MS Excel spreadsheet or CSV file.

CellProfiler

CellProfiler [28] is a software used to study and quantify cell images. It was made by Broad Institute and it has open-source code. It is user-friendly and is focused on the least experienced biologists in computer vision and programming. It is already used in different types of cells with diverse characteristics with success.

Chapter 3

A tool for oncocyte cell identification

The development of the work was done on a Windows 7 Home Premium, using the software Visual Studio 2013, Aperio ImageScope and libraries (imgproc, core, ml, flann) functions from OpenCV [12].

The Data Base used is TCGA Data Portal [29]. It is a public community resource project that any researcher can use and has data from more than 100 cases of a specific cancer type. The majority of the tumors are analyzed and have information available, such as gene and miRNA (micro RNA) expression, DNA sequence/mutation analysis and copy number variation. TCGA users can use the information freely and publish findings using the TCGA data. The images from TCGA have great quantity of information, regarding number and quality of that information. For this project, only four images were used because they are too big and have all the information needed for the process. **Figure 10** shows examples of the images found in the data base.

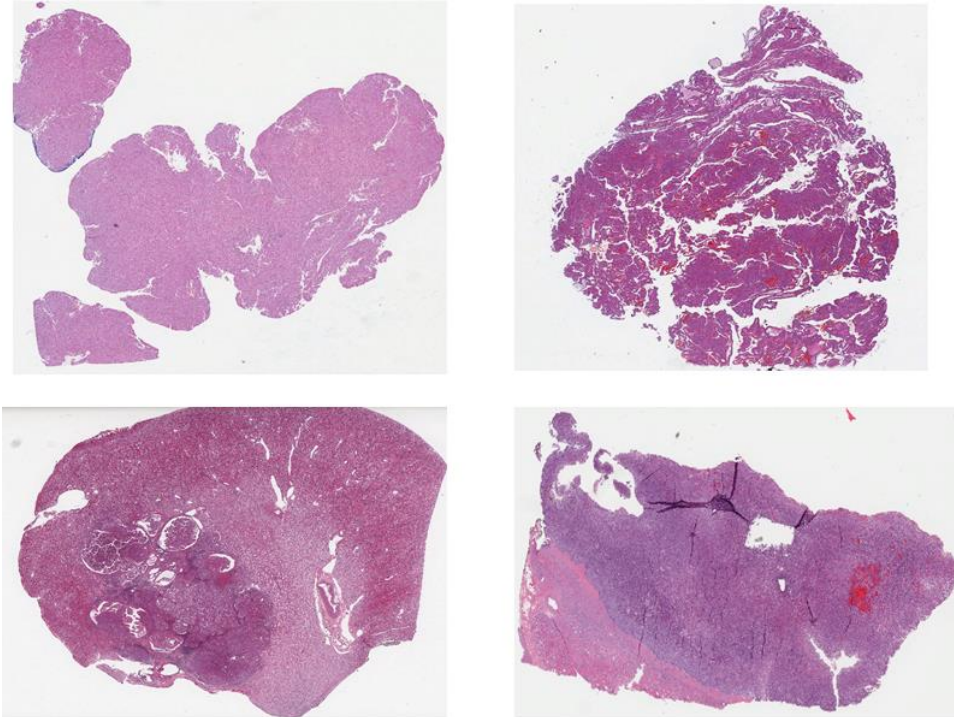


Figure 10 - Examples of images from the TCGA data base

3.1- Development

We now describe, the algorithms used.

Converting the images

The images from the data base, previously mentioned, were presented in the ScanScope Virtual Slides (SVS) format, so we had to convert them to Tagged Image File Format (TIFF) or Joint Photographic Experts Group (JPEG) because there are numerous libraries and functions that do not work with that format. To perform this task we had to use the program Aperio ImageScope, open the image, with the right click get the extraction region tool and then select the TIFF conversion. It was decided to use the TIFF format because it contains more information and more detail of the image than the other outputs.

Dividing the images

The image needs to be divided so it can be processed by the OpenCV functions because it has too much information to be processed. Due to the size of the image it is still not possible to open or use C++ functions that can open and cut the images. The images are divided using the function Rect and copyTo from the OpenCV.

The code is the following:

Arguments:
Width, height, initial position on x (x), initial position on y (y), original image
Code:
<pre>Rect window(x, y, width, height); Mat image_window = original_image(window); Image_window.copyTo(cropped_image); Imwrite("croppedimage.tiff", cropped_image);</pre>
Result:
We have the window to work saved on croppedimage.

There are some rules to divide the image. The goal is to be able to identify all the cells, it is not possible to divide in the same places because some of them could be cut and, then, not identified. So the division of the images must have 5% of the other image, previously cropped, so it is guaranteed that all the cells can be identified.

Segmentation of the cells

With the new smaller images we can begin the procedure to identify the cells. The images are transformed to grey scale to save time in the process time without losing information, by using the function `cvtColor(original_image, gray_image, CV_BGR2GRAY)`.

After converting the image to grey scale, we apply a filter represented in **Figure 11** to get high contrasts. Doing this, it is possible to differentiate the objects from the background easily.

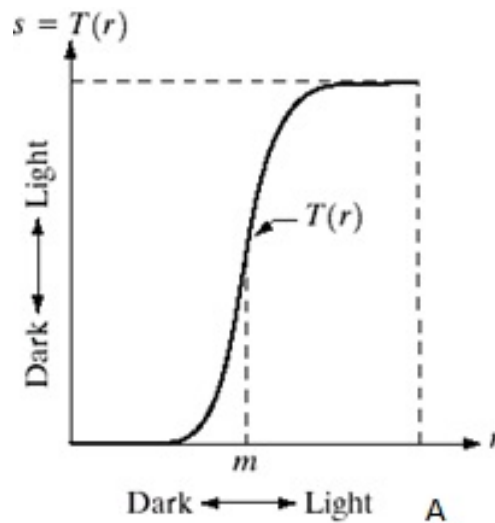


Figure 11 - Gray level transformation function for high contrast enhancement

This process made the objects darker and easier to be identified.

Afterwards, it is vital to detect the objects (cells or not) present on the image. In order to perform this task, we process the function Canny() [30] that will make the image clearer and just represent the edges of all the objects found in the image. The Canny algorithm is considered to be the optimal edge detection algorithm so the output is a binary image with only the edges from the objects present in the image.

After that, some filters are applied to reduce the noise like min/max and average, erasing the dots that are too small to be considered cells. At this point the image contains objects that can be cells or not.

If some borders are still together, it is used the function watershed() that uses the watershed algorithm.

Afterwards, a function is used to determine the location of the borders found in the last step. That function is called findContours() [12] and the code is as follows.

Arguments:
canny_input (image to be analyzed), contours_output (vector with vector of points to each contour), original_image, CV_RETR_EXTERNAL (only retrieves the outer contours), method CV_CHAIN_APPROX_NONE (stores all the points from the contour)
Code:

```

findContours(canny_input, contours_output, hierarchy, CV_RETR_EXTERNAL,
CV_CHAIN_APPROX_NONE, Point(0,0));
for( int i = 0; i< contours_output.size(); i++ )
{
    Scalar color = Scalar( rng.uniform(0, 255), rng.uniform(0,255), rng.uniform(0,255) );
drawContours(original_image, contours_output, color, 2, 8, hierarchy, 0 , Point());
}

```

Result:

We have the contours on contours_output and they are drawn on top of the original_image with different colors so they can be easily seen.

With this output image, the objects can be classified as cell or not cell by a pathologist. If they are selected as cell they stay in the vector as cells, if not, the non-cells are negative data to the machine learning process. The next step is extracting characteristics.

Extracting characteristics from the cells

On this part of the code it is proposed to get the largest number of information in each object/cell. We use cv2.contourArea() and cv2.arcLength() to get information about the area and perimeter of the cells. Other characteristics can be gathered by processing the information from the cells within the contour and pixels near. These characteristics are stored as well as the position of the cells to train a machine learning algorithm and decide which are the best characteristics to use to detect cells.

3.2 - Results

Converting and dividing images

The conversion of the images from SVS format to TIFF format was 100% successful using the software Aperio ImageScope. However, the program is not the best because it crashes 2 in 5 times when it is converting and sometimes does not open the extracting region tool.

Using C++ functions or libraries it was not possible to divide the images that resulted from the previous step. This happened because they were still too big (more than 4 GB) and the software or system could not open the image with that size. OpenCV libraries, Matlab and also Paint were other tried options and none of these options worked. After reading some literature it was discovered that the processor cannot open images bigger than 2 GB neither crop them without opening them. That way is not possible to perform the next steps on the original images.

Segmentation of the cells and extracting their characteristics

The following results are not made on the original images and are just possible examples of the code explained before.

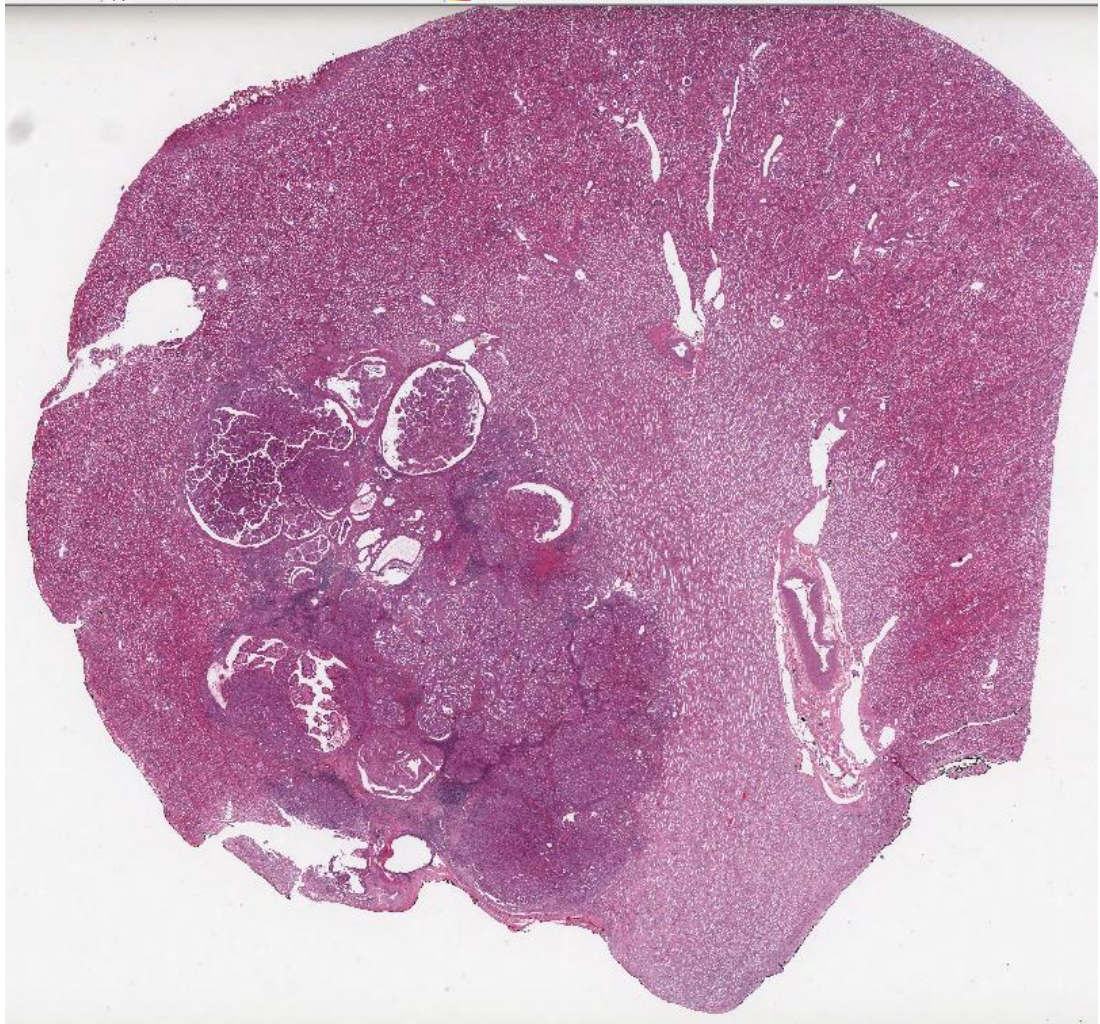


Figure 12 - Image from the TCGA data base (from [27]) with the reference “TCGA-BQ-7053-01Z-00-DX1”

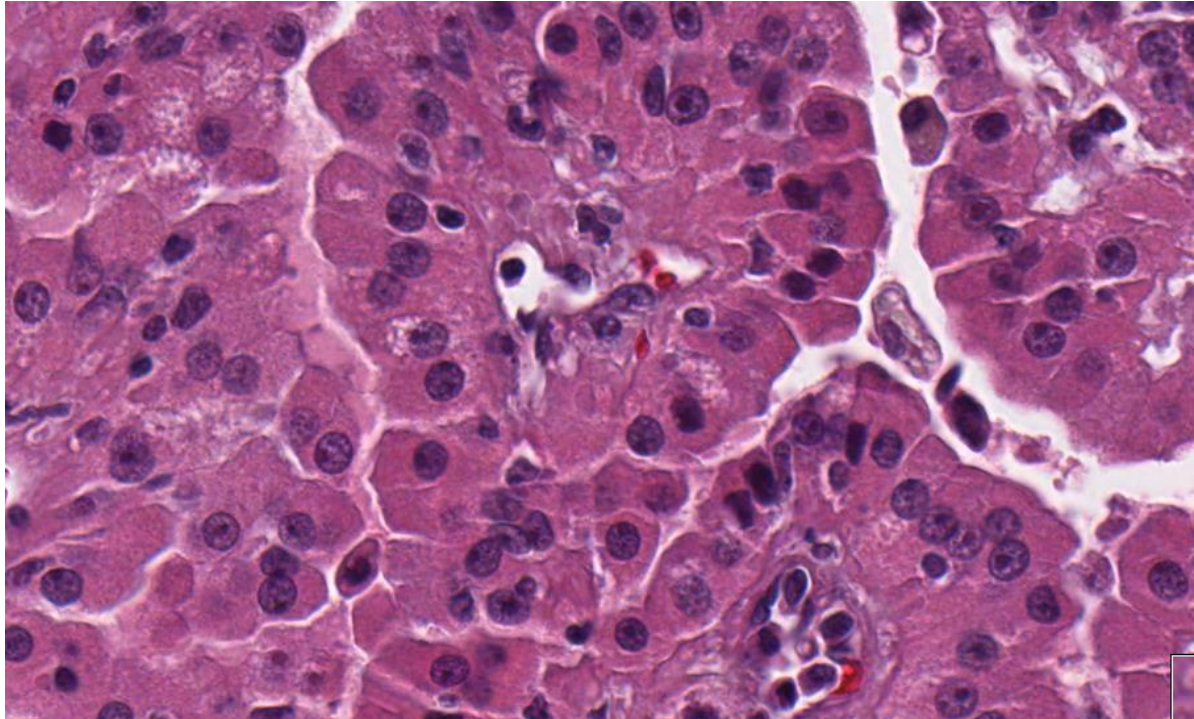


Figure 13 - Magnified part of the **Figure 12**

Figure 13 represents the original image, it is a small part of one image (**Figure 12**) from the TCGA data base with the reference TCGA-BQ-7053-01Z-00-DX1. The **Figure 14** is the result of applying the function to put the image on gray scale.

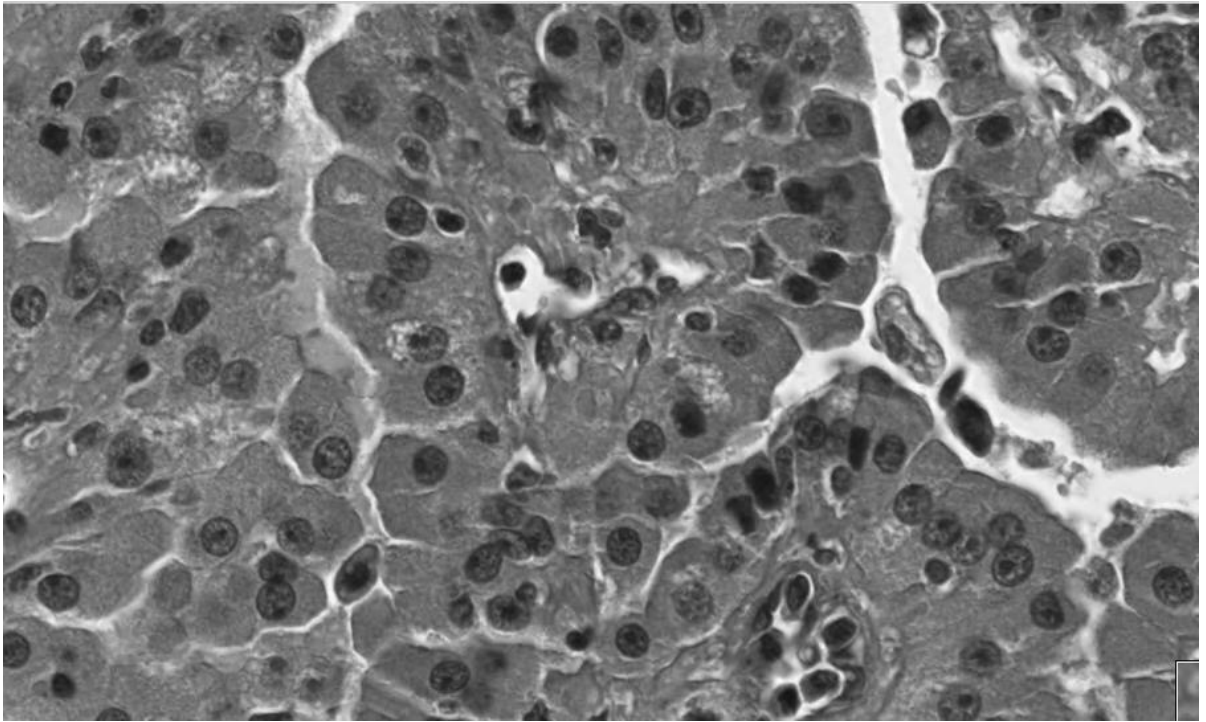


Figure 14 - Result of converting **Figure 13** to gray scale

At this point, a function is applied to get higher contrast between the scales of gray. On **Figure 15** it is possible to see the probable cells at darker color.

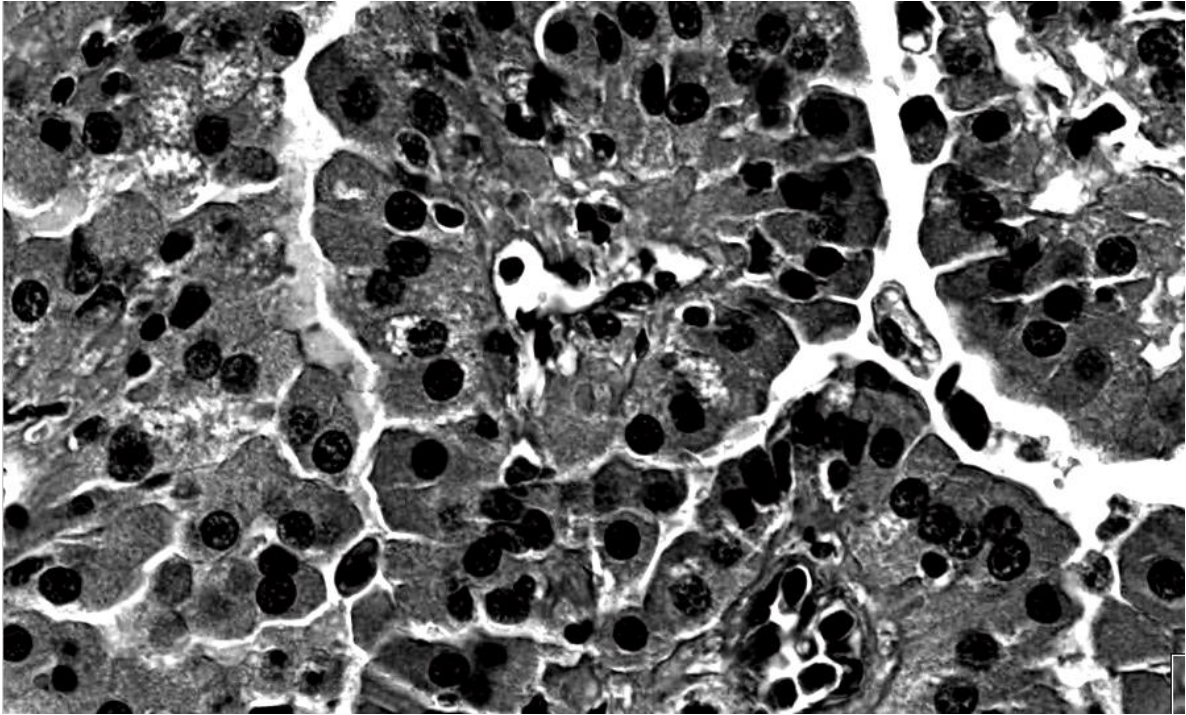


Figure 15 - Result of applying high contrast functions to Figure 14

The Figure 16 shows the possible cells with noise.

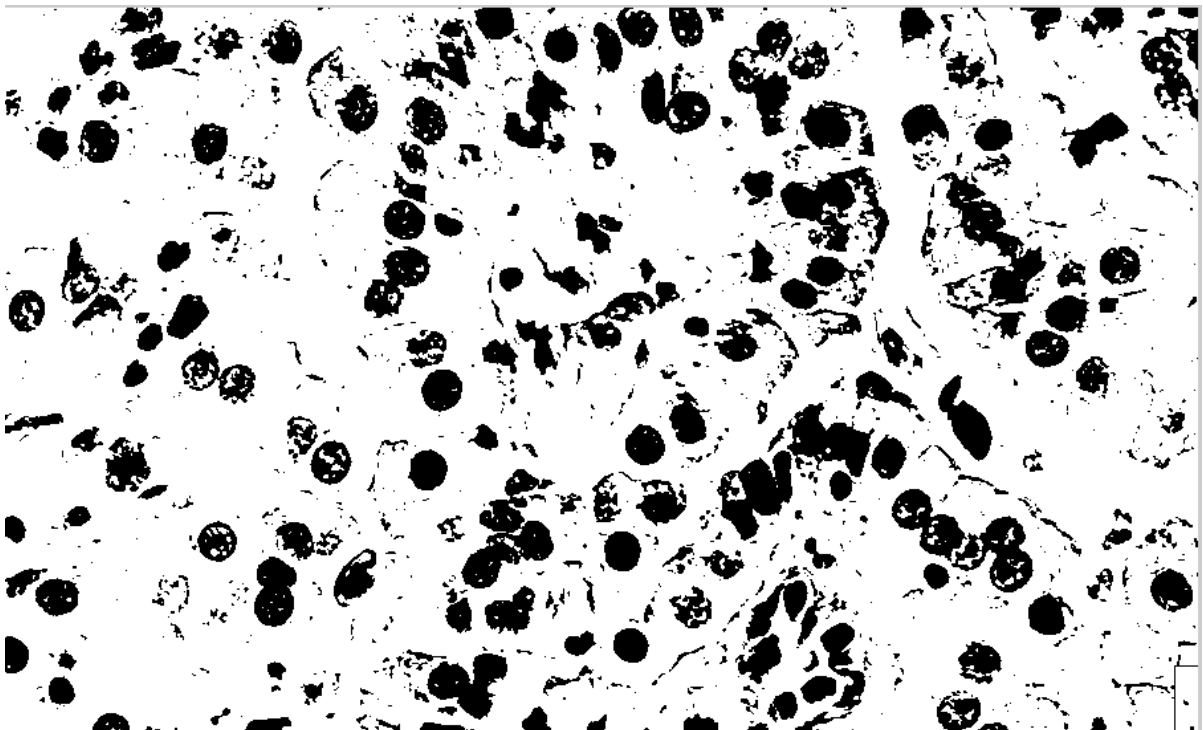


Figure 16 - Binary image from the possible cells

Using the canny function we get Figure 17.

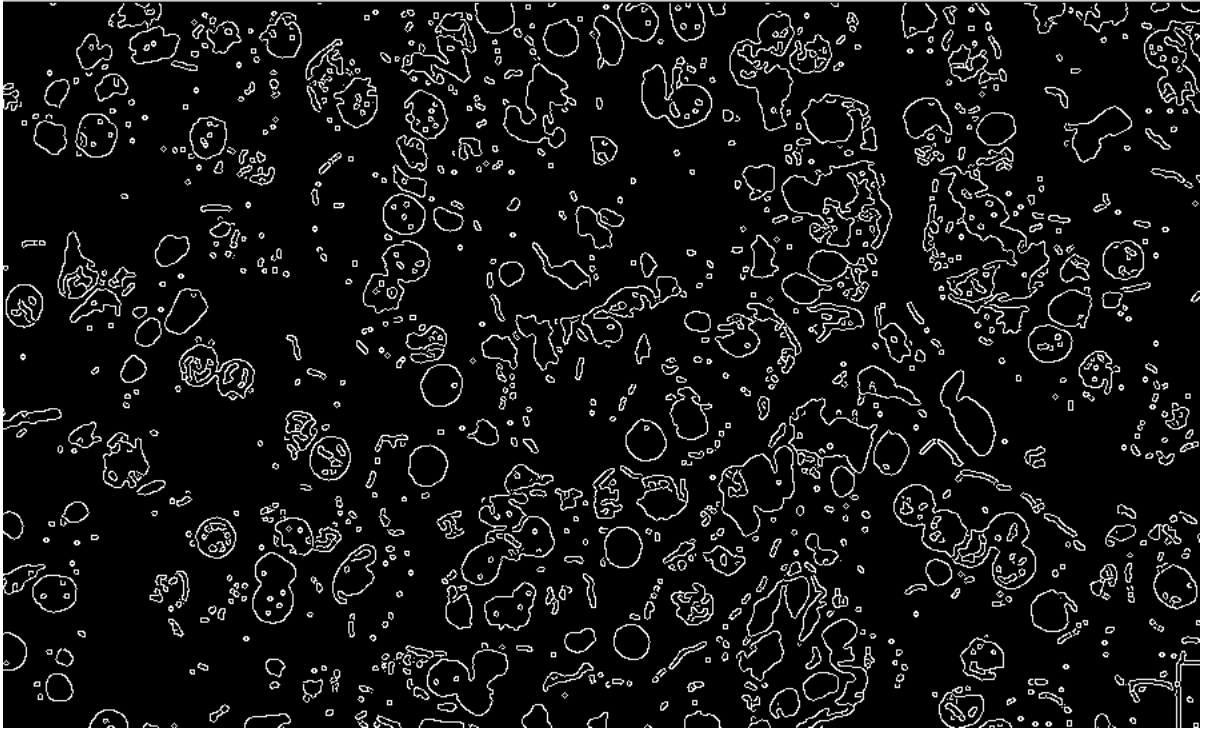


Figure 17 - Borders of the objects found in Figure 16

Now, all the objects represented by the borders have been identified, the smaller ones can be considered noise so they are removed from the data by computing that items with an area smaller than a threshold, e.g. the media of all the items, are removed. The others proceed to next step, where they are masked in the original image, labeled and the characteristics such as the area, perimeter, color from the original image, roundness, etc. are gathered.

Afterwards, the objects are classified as cells or not by the pathologist. That way they can be used on the machine learning functions like truth examples and the objects that are not classified as cells as false.

On this chapter the objectives were accomplished but the images used were only examples and not the originals. We also gathered a good number of characteristics from the cells that can be used by the data mining tools.

Chapter 4

Machine learning

The development of the work was done on a Windows 7 Home Premium, using the software Weka.

The data used by the machine learning algorithms is the data previous classified by the pathologist. The dataset⁴ was composed by 18 072 classified instances, being split in equal parts, 9032 labeled oncocytic and 9032 labeled others. We have use the hold-out method to estimate the quality of the classifiers, so the data was split in two subsets the first to train, with 12492 instances, and the second to be used as test set with 5 580 instances.

4.1 - Development

The data was then classified by six classification algorithms so they can be compared and the results analyzed:

- ***K*-nearest Neighbor**
- **Naïve Bayes**
- **Decision Trees - J48**
- **Clustering**
- **SVM**
- **Decorate**

Two sets of experiments were done using different number of attributes. The first experiment used only five attributes (area, length, etc.) the second on used 46 attributes

⁴ Since we have not finish the software tool we use a dataset provided by Tiago Mota (student from FEUP that worked seriously on this topic)

with the first four and more (min, max and mean of the intensity of grey, blue, red and green pixels, etc.).

4.2 - Results

The **Figure 18** shows the results from the first experiment with 5 attributes.

5_attributes	Accuracy	Kappa	F-Measure	Time taken
K-nearest Neighbor	98%	0.969	0.985	0.05 s
Naive Bayes	79%	0.587	0.793	0.02 s
Decision Trees - J48	93%	0.858	0.929	0.31 s
Clustering	59%	0.175	0.584	0.25 s
SVM	80%	0.607	0.803	0.35 s
Decorate	95%	0.897	0.948	59.91 s

Figure 18. Results of the classification with 5 attributes

It is clear that the classifiers with best results were the *K*-nearest neighbor, Decorate and Decision trees with an accuracy ranging from 93% to 98%. Comparing these three classifiers using the other measures the *K*-nearest Neighbor can be considered the best classifier because his kappa is considered almost perfect [21] and his F-measure is higher than the others, the time taken is the lowest of the three. So we get the best result with the best time using *K*-nearest Neighbor classifier.

Naïve Bayes and SVM classifiers have an accuracy of 79-80% but their kappa is moderate. The Clustering classifier gets a low accuracy of 59% and his kappa and F-measure are very low comparing to the others classifiers, so it can be considered to be useless for this problem.

Next it is shown, in **Figure 19**, the results of all the methods with 46 attributes.

46_attributes	Accuracy	Kappa	F-Measure	Time taken
K-nearest Neighbor	99%	0.97	0.986	0.04 s
Naive Bayes	69%	0.38	0.676	0.15 s
Decision Trees - J48	98%	0.96	0.980	2.21 s
Clustering	72%	0.43	0.713	3.18 s
SVM	86%	0.73	0.863	3.49 s
Decorate	99%	0.98	0.989	427.67 s

Figure 19. Results of the classification with 46 attributes

Regarding the classification with 46 attributes we get better results overall than with only five attributes. Only Naive Bayes method gets worst results in all measures, and the time taken is worst because there is more information to be processed.

The *K*-nearest Neighbor and Decorate methods have an accuracy of 99%, the kappa (considered almost perfect [21]) and F-measure are very good. The Decorate has better results but takes more than three minutes and the *K*-nearest Neighbor takes only 0,04 seconds so it is considered the best classification method again.

The Decision trees algorithm has good accuracy, close to 99%, his kappa is considered as almost perfect and his F-measure is higher than 0,96 so it can be considered good a method as well.

As it was said before the Naïve Bayes has the worst results and the clustering gets better results than with only five attributes and it is considered a moderate result regarding the kappa measure but is still one of the worst methods. The other (SVM) is considered a substantial result regarding the kappa and has F-measure higher than 0,85, meaning that is also a good method.

Chapter 5

Conclusion and future work

5.1 - Conclusion

The goals could not be achieved with the original images and, instead, they were made with parts of them. In view of this fact, it can be possible to use these procedures to get a favorable result on identifying the objects from the images.

We have accomplished the objective of converting the images from SVS format to TIFF format. Using the image on a gray scale, applying the high contrast function followed by the canny procedure is a good way of detecting objects. The features extracted from that objects can be used on machine learning functions so the software can detect automatically cells in the given images.

We have used six machine learning algorithms and it is clear that the *K*-nearest Neighbor has the best results with the two tests. Decorate and Decision trees algorithms have good results and can also be used to get a good classification on this problem. Based on this results the clustering method is considered to be the worst one. The others have satisfactory results.

5.2 - Future work

To improve on this work one of the big problems is dividing the images or opening them. The only approach to do that is the evolution of the processors on the computers or by trying to make a library that can only open parts of images, like selecting prior to opening the location and size of a window.

Regarding the procedures, it is suggested to apply them in more images and consult a pathologist to confirm the best way on identification without losing information from the images.

Related to the machine learning part it is possible to use more algorithms and bigger datasets so the results can improve.

References

- [1]. Máximo, V. and Lima, J. and Soares, P. 2009. "Mitochondria and cancer." *Virchows Archiv: an international journal of pathology* no. 454 (5):481-495
- [2]. Pereira, L. and Soares, P. and Máximo, V. 2012 "Somatic mitochondrial DNA mutations in cancer escape purifying selection and high pathogenicity mutations lead to the oncogenic phenotype: pathogenicity analysis of reported somatic mtDNA mutations in tumors." *BMC cancer* no. 12 (1):53
- [3]. "Berkeley Cancer Morphometric Data"
<http://tcga.lbl.gov:8080/biosig/tcgadownload.do>
- [4]. Alberts, B. and Johnson, A. and Lewis J. 2002. *Molecular Biology of the Cell*. 4th edition. New York: Garland Science
- [5]. "Cell Biology and Cancer"
http://www.learner.org/courses/biology/support/8_cancer.pdf
- [6]. Oghan, F. and Apuhan, T. and Guvey, A. "Rare Malignant Tumors of the Parotid Glands: Oncocytic Neoplasms." *Neck Dissection - Clinical Application and recent Advances* 137-148.
- [7]. Chakrabarti, I. and Basu, A. and Ghosh, N. 2012. "Oncocytic lesion of parotid gland: A dilemma for cytopathologists" *journal of Cytology* 29(1): 80-82
- [8]. Asa, S. L. 2004. "My approach to oncocytic tumours of the thyroid" *J Clin Pathol* 57: 225-232
- [9]. "Capítulo 1 da componente teórica de Sistemas Baseados em Visão"
https://sigarra.up.pt/feup/pt/conteudos_service.conteudos_cont?pct_id=116901&pv_cod=27iaaPjaEXjX
- [10]. Tschumperlé, D. "The CImg Library, C++ template image processing toolkit."
<http://cimg.sourceforge.net>
- [11]. Schalnat, G. E. and Dilger, A. and Bowler, J. and Randers-Pehrson, G. "libpng, the official PNG reference library." <http://www.libpng.org/pub/png/libpng.html>
- [12]. Bradski, G. 2000. *Dr. Dobb's Journal of Software Tools*
<http://docs.opencv.org/index.html>
- [13]. Ecole des mines de Paris "Camellia Image Processing & Computer Vision library."
<http://camellia.sourceforge.net/index.html>
- [14]. Han, J. and Kamber, M. 2000. "Data Mining: Concepts and Techniques" Morgan Kaufmann Publishers
- [16]. Magerman, David M.. *Statistical Decision-Tree Models for Parsing*. Cambridge, MA 02138, USA
- [17]. Asa Ben-Hur e Jason Weston. *A User's Guide to Support Vector Machines*. Colorado State University, Princeton, NJ 08540 USA.
- [18]. Altman, N. S. 1992. "An introduction to kernel and nearest-neighbor nonparametric regression". *The American Statistician* 46 (3): 175-185
- [20]. P. Melville, R. J. Mooney: *Constructing Diverse Classifier Ensembles Using Artificial Training Examples*. In: *Eighteenth International Joint Conference on Artificial Intelligence*, 505-510, 2003.
- [21]. Hand, D. J.; Yu, K. (2001). "Idiot's Bayes – not so stupid after all?". *International Statistical Review* 69 (3): 385-399

- [22]. Christopher M. Bishop (2006). *Pattern Recognition and Machine Learning*. Springer. p. 205. "In the terminology of statistics, this model is known as *logistic regression*, although it should be emphasized that this is a model for classification rather than regression."
- [23]. Landis, J.R.; Koch, G.G. (1977). "The measurement of observer agreement for categorical data". *Biometrics* **33** (1): 159-174.
- [24]. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); *The WEKA Data Mining Software: An Update*; SIGKDD Explorations, Volume 11, Issue 1.
- [25]. Mierswa, I. and Wurst, M. and Klinkenberg, R. and Scholz, M. and Euler, T. 2006. "Yale: Rapid prototyping for complex data mining tasks" Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining 935-940
- [26]. Berthold, M. R. and Cebron, N. and Dill, F. and Gabriel, T. R. and Kotter, T. and Meinl, T. and Ohl, P. and Sieb, C. and Thiel, K. and Wiswedel, B. 2007. "KNIME: The Konstanz Information Miner" *Studies in Classification, Data Analysis, and Knowledge Organization*
- [27]. Demsar, J. and Curk, T. and Erjavec, A. and Gorup, Č. and Hočevar, T. and Milutinovič, M. and Možina, M. and Polajnar, M. and Toplak, M. and Starič, A. and Štajdohar, M. and Umek, L. and Žagar, L. and Žbontar, J. and Žitnik, M. and Zupan, B. 2013. "Orange: Data Mining Toolbox in Python" *Journal of Machine Learning Research* no 14: 2349-2353
- [28]. Zhou, J. and Lamichhane, S. and Sterne, G. and Ye, B. and Peng, H. 2013. "BIOCAT: a pattern recognition platform for customizable biological image classification and annotation" *BMC Bioinformatics* no. 1(14)
- [29]. "CellNote" <http://cellnote.up.pt/>
- [30]. Jones, T. R. and Kang, I. H. and Wheeler, D. B. and Lindquist, R. and Papallo, A. and Sabatini, D. M. and Golland, P. and Carpenter, A. E. 2008. "CellProfiler Analyst: data exploration and analysis software for complex image-based screens." *BMC bioinformatics* <http://www.cellprofiler.org/>
- [31]. "The Cancer Genome Atlas" <https://tcga-data.nci.nih.gov/tcga/>
- [32]. Canny, J. 1986 "A Computational Approach To Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*", 8(6):679-698, 1986